table defines one particular state transition, the conversion matrix has three entries, one at the intersection of current state row 1001 and next state column 1101, row 1000 and column 1101, and row 100X and column 1101.

The paragraph at Page 24, Lines 20-23:

At block 1110, the process receives a data structure representing the behavior of a sequential circuit element written in a specific HDL format. As discussed above, the data structure may have come, for instance, from a library of HDL elements from a particular integrated circuit fabricator based on a particular fabrication technology.

IN THE CLAIMS

Please replace the following amended claims:

1. (Once Amended) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

7. (Once Amended) A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition; and

wherein states for individual signals comprise 0, 1, and X.

B2 cont'd

8. A method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

A6

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein the conversion matrix is organized into a plurality of current states of the circuit element and corresponding next states of the circuit element, and wherein determining the generic HDL register and the plurality of generic HDL input logic comprises:

classifying each next state of the conversion matrix into one of a plurality of sets of states based on predefined criteria;

selecting either an edge sensitive HDL primitive or a level sensitive HDL primitive for the generic HDL register based on selected ones of the plurality of sets of states;

selecting a particular set of functions based on the generic HDL register selected, each function of the particular set of functions corresponding to an input to the generic HDL register; and

evaluating the particular set of functions to determine the plurality of generic HDL input logic.

12. (Once Amended) The method of claim 9 wherein the particular set of functions for the edge sensitive HDL primitive comprises:

Fset = L01 with do not cares for a union of L11, LUR01, and LUR11;

Frst = L10 with do not cares for a union of L00, LUR10, and LUR00;

Fclen = a single common input identified in sets E01 and E10;

Ftmpa = selected next states from a union of E01, E11, and L11;

Ftmpb = Ftmpa with do not cares for selected next states from a union of EUR01, EUR11, and LUR11;

Fd = Ftmpb with do not cares for a union of Fset and Frst, wherein the single common input identified for Fclen is ignored, and wherein the selected next states are edge states having a same edge transition as the single common input identified for Fclen

13. (Once Amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = L01 with do not cares for L11;

Frst = L10 with do not cares for L00;

Fclen = 0; and

Fd = 0.

14. (Once Amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = 0;

Frst = 0;

Fclen = (L01 with do not cares for L11) union with (L10 with do not cares for L00); and

Fd = L01 with do not cares for L11.

15. (Once Amended) The method of claim 9 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = a single input identified from the connectivity matrix for which there is no current state for which the output Q+ of the corresponding next state is one;

Frst = a single input identified from the connectivity matrix for which there is no current state for which the output Q+ of the corresponding next state is zero;

Ftmp1 = L01 with do not cares for a union of L11, LUR01, and LUR11;

Ftmp2 = L10 with do not cares for a union of L00, LUR10, and LUR00;

Fclen = a union of Ftmp1 and Ftmp2 with do not cares for a union of Fset and Frst; and

Fd = Ftmp1 with do not cares for an inverted Fclen.

16. (Once Amended) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.

22. (Once Amended) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein generating the conversion matrix comprises:

identifying input signals and an output signal of the circuit element from the data structure;

evaluating state transitions for the circuit element by evaluating the data structure for a next output signal for each transition of the input signals and for each current output signal; and

populating entries of the conversion matrix for each state transition; and

wherein states for individual signals comprise 0, 1, and X.

23. (Once Amended) A machine readable medium having stored thereon machine executable instructions to implement a method for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the method comprising:

receiving the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generating a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format;

determining a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix;

wherein the conversion matrix is organized into a plurality of current states of the circuit element and corresponding next states of the circuit element, and wherein determining the generic HDL register and the plurality of generic HDL input logic comprises:

classifying each next state of the conversion matrix into one of a plurality of sets of states based on predefined criteria;

selecting either an edge sensitive HDL primitive or a level sensitive HDL primitive for the generic HDL register based on selected ones of the plurality of sets of states;

selecting a particular set of functions based on the generic HDL register selected, each function of the particular set of functions corresponding to an input to the generic HDL register; and

evaluating the particular set of functions to determine the plurality of generic HDL input logic.

---

27. (Once Amended) The machine readable medium of claim 24 wherein the particular set of functions for the edge sensitive HDL primitive comprises:

Fset = L01 with do not cares for a union of L11, LUR01, and LUR11;

Frst = L10 with do not cares for a union of L00, LUR10, and LUR00;

Fclen = a single common input identified in sets E01 and E10;

Ftmpa = selected next states from a union of E01, E11, and L11;

Ftmpb = Ftmpa with do not cares for selected next states from a union of EUR01, EUR11, and LUR11;

Fd = Ftmpb with do not cares for a union of Fset and Frst, wherein the single common input identified for Fclen is ignored, and wherein the selected next states are edge states having a same edge transition as the single common input identified for Fclen

28. (Once Amended) The machine readable medium of claim 24 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = L01 with do not cares for L11;

Frst = L10 with do not cares for L00;

Fclen = 0; and

Fd = 0.

29. (Once Amended) The machine readable medium of claim 24 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = 0;

Frst = 0;

Fclen = (L01 with do not cares for L11) union with (L10 with do not cares for L00); and

Fd = L01 with do not cares for L11.

30. (Once Amended) The machine readable medium of claim 16 wherein the particular set of functions for the level sensitive HDL primitive comprises:

Fset = a single input identified from the connectivity matrix for which there is no current state for which the output Q+ of the corresponding next state is one;

Frst = a single input identified from the connectivity matrix for which there is no current state for which the output Q+ of the corresponding next state is zero;

Ftmp1 = L01 with do not cares for a union of L11, LUR01, and LUR11;

Ftmp2 = L10 with do not cares for a union of L00, LUR10, and LUR00;

Fclen = a union of Ftmp1 and Ftmp2 with do not cares for a union of Fset and Frst; and

Fd = Ftmp1 with do not cares for an inverted Fclen.

31. (Once Amended) An apparatus for converting a data structure from a specific format in a hardware description language (HDL) to generic HDL elements, the apparatus comprising:

a processor; and

a machine readable storage medium storing thereon machine executable instructions, the processor to execute the machine executable instructions to

receive the data structure representing a behavior of a circuit element, said circuit element being sequential and said data structure being defined using the specific format;

generate a conversion matrix from the data structure, said conversion matrix to represent the behavior of the circuit element in a generic format; and

determine a generic HDL register and a plurality of generic HDL input logic for the generic HDL register to replicate the behavior represented by the data structure based on the conversion matrix.